📊 Project Report: Credit Score Risk Classifier

# 🎯 Objective

The aim of this project is to build a machine learning model that classifies customers into credit risk categories—Low, Medium, and High—based on financial data. Additionally, the project demonstrates cloud-based storage integration using AWS S3, where input and output files are uploaded after local processing.

# 📁 Dataset Description

The dataset is synthetically generated and includes:  
- Age  
- Income  
- Loan Amount  
- Credit Score  
Credit scores are used to classify risk levels as:  
- High risk: score < 580  
- Medium risk: 580 ≤ score < 670  
- Low risk: score ≥ 670

# 🧰 Technologies & Tools

- Python

- pandas, numpy

- scikit-learn

- boto3 (AWS SDK for Python)

- Jupyter Notebook

- AWS S3 (Simple Storage Service)

# ⚙️ Approach

1. Generate synthetic data for credit analysis.  
2. Label the risk level using business logic.  
3. Train a Random Forest Classifier.  
4. Evaluate model performance using classification metrics.  
5. Save input data and prediction results locally.  
6. Upload the files to AWS S3 using boto3 client.

# 🧠 Code Highlights

- `label\_risk(score)`: Function to classify risk level.

- `RandomForestClassifier()`: Model used for training.

- `classification\_report()`: To evaluate the classifier.

- `boto3.client('s3', region\_name='us-east-1')`: Used for AWS S3 access.

- `s3.upload\_file(...)`: Uploads files to respective paths in the S3 bucket.

# 📤 Output Summary

✅ Input File: `data/credit\_data.csv`

✅ Output File: `output/risk\_predictions.csv`

✅ Both files are successfully uploaded to:

- `s3://shrushti-credit-bucket/input/credit\_data.csv`

- `s3://shrushti-credit-bucket/output/risk\_predictions.csv`

# ✅ Conclusion

This project successfully demonstrates the application of machine learning to a real-world-style credit risk problem and integrates cloud storage for seamless data pipeline workflows. It’s a production-ready prototype that showcases the ability to build, deploy, and scale ML-based financial tools.